531 Rec'd PCT/ 19 APR 2001

SUBSTITUTE FORM PTO-1390 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER
13109-002001

U.S. APPLICATION NO. (If Known, see 37 CFR 1.5) 09/830049

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/GB99/03438	18 October 1999	19 October 1998

TITLE OF INVENTION
PARALLEL PROCESSOR FOR MOTION ESTIMATOR

APPLICANT(S) FOR DO/EO/US
SERGEY ARTAMONOV, VLADIMIR KOZLOV, YURY ZAUTLIVETER, ELENA FISCHENKO

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to promptly begin national examination procedures (35 U.S.C. 371(f)).
4. ☐ The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. ☒ is attached hereto (required only if not communicated by the International Bureau).
   b. ☐ has been communicated by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☐ An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. ☐ are attached hereto (required only if not communicated by the International Bureau).
   b. ☐ have been communicated by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ An English language translation of amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. ☐ An English language translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11 to 16 below concern other documents or information included:

11. ☐ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☐ A FIRST preliminary amendment.
    ☐ A SECOND or SUBSEQUENT preliminary amendment.
14. ☐ A substitute specification.
15. ☐ A change of power of attorney and/or address letter.
16. ☐ Other items or information:
    ☒ express mail declaration
    ☒ return postcard
    ☐
    ☐
    ☐

CERTIFICATE OF MAILING BY EXPRESS MAIL Express Mail Label No. EL688265324US

I hereby certify under 37 CFR § 1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231

19 April 2001
Date of Deposit

Signature

Gil Vargas
Typed Name of Person Signing

JC02 Rec'd PCT/PTO 19 APR 2001

U.S. APPLICATION NO. (IF KNOWN) 09/830049	INTERNATIONAL APPLICATION NO. PCT/GB99/03438	ATTORNEY'S DOCKET NUMBER 13109-002001

17. ☐ The following fees are submitted:

CALCULATIONS PTO USE ONLY

Basic National Fee (37 CFR 1.492(a)(1)- (5)):

Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO..... **$1000**

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO **$860**

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO..... **$710**

International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4)..... **$690**

International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4)..... **$100**

			ENTER APPROPRIATE BASIC FEE AMOUNT =	$860.00	
Surcharge of **$130** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				$0.00	
Claims	Number Filed	Number Extra	Rate		
Total Claims	- 20 =		x **$18**	$0.00	
Independent Claims	- 3 =		x **$80**	$0.00	
MULTIPLE DEPENDENT CLAIMS(S) (if applicable)			+ **$270**	$0.00	
			TOTAL OF ABOVE CALCULATIONS =	$860.00	
☐ Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above are reduced by 1/2.				$430.00	
			SUBTOTAL =	$430.00	
Processing fee of **$130** for furnishing the English Translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492(f))				$0.00	
			TOTAL NATIONAL FEE =	$430.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00** per property +				$0.00	
			TOTAL FEES ENCLOSED =	$430.00	
				Amount to be refunded:	$
				Charged:	$

a. ☒ A check in the amount of $430.00 to cover the above fees is enclosed.
b. ☐ Please charge my Deposit Account No. 06-1050 in the amount of $0.00 to cover the above fees. A duplicate copy of this sheet is enclosed.
c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 06-1050. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Scott C. Harris
FISH & RICHARDSON P.C.
4350 La Jolla Village Drive, Suite 500
San Diego, CA 92122
(858) 678-5070 *phone*
(858) 678-5099 *facsimile*

SIGNATURE : Reg No 44,631

NAME Scott C. Harris

REGISTRATION NUMBER 32,030

PARALLEL PROCESSOR FOR MOTION ESTIMATOR

This invention relates to video encoding and decoding, and in particular to the calculation of motion vectors in a video compression system such as MPEG-2.

The MPEG-2 video standard is defined in ISO/IEC 13818-2 and is based on elimination of redundant video data to enable high quality picture information to be transmitted over a relatively narrow bandwidth channel. Video compression is achieved in a number of separate ways including intra-frame coding and inter-frame coding. Intra-frame coding reduces video data first by quantising discrete cosine transfer (DCT) coefficients of spatial data. The image to be coded is divided into a number of macroblocks each of 16 x 16 pixels and a different quantizing scale may be defined for each macroblock. Following quantisation lossless data reduction is applied by using Variable Length Coding (VLC) and Run Length Coding (RLC) to reduce the number of bits required to encode common patterns and frequently occurring values. The image to be encoded is divided into a number of macroblocks each of 8 x 8 pixels. Variable Length Coding and Run Length Coding is performed on 8 x 8 pixel blocks using a zigzag pattern to maximised redundancy.

Inter-frame compression seeks to eliminate information which is redundant by virtue of it having been present in a past, or future image defined as an anchor frame. The anchor frame is a full resolution, full data picture. As the image will often contain portions which are moving from frame to frame, motion vectors are used to predict a present frame from an anchor frame. Motion vectors are assigned at a macroblock level and the predicted frame is subtracted from the actual frame to form a difference frame which has a much lower information

context than the actual frame. The content of the difference frame will depend on the accuracy of the predicted frame. The predicted frame is developed from a IDCT quantised, decoded picture.

Inter-frame prediction may be based solely on forward prediction from intra-frame coded images or other forward predicted frames, or be bi-directionally predicted from both a previous and a future intra-frame coded or forward predicted frame. Bidirectional coding necessarily means that the video input order must be changed so that the past and the forward anchor frames are known.

The MPEG-2 standard provides a number of defined system configurations which are represented as levels and profiles as shown in table 1 below.

LEVEL/ PROFILE	SIMPLE	MAIN	SNR	SPATIAL	HIGH
HIGH		1920x1152 80 Mb/s			1920x1152 100 Mb/s
HIGH 1400		1440x1152 60 Mb/s		1440x1152 60 Mb/s	1440x1152 80 Mb/s
MAIN	720x576 15 Mb/s	720x576 15 Mb/s	720x576 15 Mb/s		720/576 20 Mb/2
LOW		352x288 4 Mb/s	352x288 4 Mb/s		

The MPEG-2 standard is designed to be scalable, that is decoders and encoders do not need to be of comparable quality to work together. It is desirable to design motion estimation processors which use corresponding VLSI technologies for the corresponding MPEG profiles. Where possible it is desirable that the processors should be on a single chip. However, where this is not yet possible, for the highest profiles and levels, it is desirable to be

able to operate a plurality of motion estimation processors in parallel.

In addition to ensure that the maximum degree of video data compression can be achieved, within the confines of the MPEG-2 standard, it is desirable to be able to search the whole of the frame with a half-pixel accuracy.

Computationally, the calculations on motion vectors is the hardest operation in coding video to the MPEG standard. The processes is illustrated in figure 1 in which the forward anchor frame is identified by the reference numeral 10, the backward anchor frame by the numeral 14 and the current frame by 12. In figure 1 it can be seen that a given macroblock 16 is in a different position in each of the three frames, indicating a non-constant velocity movement.

For each of the macroblocks in the current frame 12 it is necessary to search for the matching macroblock in the full anchor frame with a half-pixel precision. The expression for the fully search algorithm for a single current frame macroblock is:

$$(Z-X,G-Y)=Arg(\min_{X,Y}[\sum_{i=1}^{N}\sum_{j=1}^{M}|B(Z+i,G+j)-I(X+i,Y+j)|]).....(1)$$

Where X,Y are the coordinates of the left upper corner of the anchor frame macroblock;

Z,G are the coordinates of the left upper corner of the current frame macro block;

(Z-X, G-Y) are the motion vector coordinates for the current macroblock being examined; and M,N are the macroblock dimensions in pixels.

Referring now to figure 2, a half pixel precision search can be understood as being a linear interpolation of adjacent pixels. Thus, in figure 2, A,B,D,E represent pixels of the original luminance matrix and h,v,c and the two unidentified points represent half-pixels.

The half pixels are calculated by the following linear interpolations:

Horizontal Interpolation	h = (A+B)/2.....(2)
Vertical Interpolation	v = (A+D)/2.....(3)
Central Interpolation	c = (A+B+D+E)/4...(4)

As motion estimation requires the vectors of a number of macroblocks to be determined, and as video information is both spatial and temporal, parallel computing techniques are ideal for motion estimation.

There are known in the art, a number of architectures which are aimed at increasing computation performance, whilst performing a full search algorithm (within the chosen search range all possible displacements are evaluated using the block matching criterion, in contrast to logarithmic, telescopic and other searches).

In papers entitled "Arrray Architectures for Block Matching Algorithms" by T. Komarek, P. Pirsch, IEEE Trans. Circuits and Systems, Vol 36, N10, Oct. 1989, pp.1301-1308, and "Parameterizable VLSI architectures for the Full-Search and Block Matching Algorithm" by L. De Vos, M. Stegherr, IEEE Trans. Circuits and Systems, Vol 36, N10, Oct 1989 pp. 1309-1316, there is described a two-dimensional systolic matrix which achieves high computational performance by a maximum degree of parallism in the performance of operations on a single anchor frame macroblock M+N. However, the architecture disclosed has the disadvantage that it only works with a given

macroblock size and is not suitable for the processing with half pixel precision. In addition, the burst pipeline latency is such that a decrease of up to 50% in computational performance is possible. Moreover, the architecture described has a high data bandwidth requirement as it has a large number of external ports for data input and output.

Various architectures have been proposed which are free from the disadvantages of the two-dimensional systolic matrix. A one-dimensional systolic matrix is disclosed in US 4,897,720 (Wu et al) and in a paper entitled "A family of VLSI designs for the Motion Compensation Block-Matching Algorithm" by Yang, Sun and Wu, IEEE Trans, Circuits and Systems, Vol 36, N10, Oct 1989, pp. 1317-1325.

This architecture is based on performing pipelined computations for a single row of pixels in a macroblock. This reduces pipeline latency and, potentially, can calculate motion vectors to half pixel precision by using four devices operating in parallel. However, the architecture has the disadvantage of a lower computational performance compared to the two-dimensional systolic matrix.

US 5,636,293 (Lin et al) discloses an architecture designed to increase the computational performance of the one-dimensional systolic matrix. A modular architecture is used which connects one-dimensional systolic matrices in tandem, allowing acceleration of calculations in the search window without increasing the number of data points. However, this architecture has the disadvantage that it does not provide half-pixel precision and computational performance is reduced as motion vectors for a single macroblock only can be searched for in the search window.

US 5,719,642 (Lee) discloses a systolic matrix with global links for anchor frame data input into the processing elements row of a single macroblock row processing architecture. In addition, increases in anchor frame data memory can achieve 100% exploitation of hardware. However, the computation performance is limited by the number of MxN processing elements which operate in parallel. In addition, the architecture of US 5,719,642 cannot calculate motion vectors with half-pixel precision.

US 5,568,203 (Lee) discloses an architecture in which the motion estimator inputs data serially into a matrix of shift registers and simultaneously loads in parallel the anchor frame pixel data into the MxN matrix of processing elements. The matrix of processing elements provides serial calculations of the full search algorithm (equation 1). Whilst this architecture has the advantage of minimising the number of input and output ports, and fully utilizes hardware resources, it cannot calculate motion vectors with half-pixel precision. In addition, computational performance is impaired as only the MxN processing elements operate in parallel.

US 5,453,799 (Yang et al) discloses a unified motion estimator which performs MPEG-2 motion vector calculations on VLSI chips operating in parallel. However, computational performance is restricted to processing a single macroblock of the current frame in the search window.

US 5,030,953 (Chiang), discloses a matrix of signal processors, consisting of M parallel groups of sub-matrixes with N parallel operating processing elements, which calculate the sum of subtractions of absolute values for a single row of macroblocks being compared. The architecture effectively utilizes hardware resources and minimises the number of I/O ports but has restricted

computational performance as it searches the motion vector of a single macroblock of the current frame and cannot calculate motion vectors with half-pixel precision.

The invention aims to overcome or ameliorate the disadvantages with the systems described above. In its broadest form, the invention provides for the simultaneous comparison of S current frame macroblocks with the nK macroblocks of the anchor frame. Preferably, K is the number of macroblocks in the area of the anchor frame with the coordinates of the left upper corner, defined with single pixel precision, 4K is the number macroblocks in the area of the anchor frame having the coordinates of the left upper corner corresponding to half-pixel precision.

More specifically, there is provided a parallel processor for estimating motion of a given portion of a current image frame with reference to a anchor frame comprising: an input for receiving current frame data; an input for receiving anchor frame data; a two-dimensional matrix of rows and columns processing elements each processing element for comparing a given area of the current frame with at least an area of the anchor frame wherein the matrix simultaneously compares S areas of the current frame with nK areas of the anchor frame, each column of processing elements simultaneously comparing one field of the current frame data with nK fields of the anchor frame, and each row of processing elements simultaneously comparing S fields of the current frame data, with n fields of the anchor frame, the matrix having dimensions of KxS and n being an integer; means for selecting from the comparison, for each area of the current frame, an area of the anchor frame corresponding to the area of the current frame; and means for outputting data identifying the selected areas of the anchor frame.

Embodiments of the invention have the advantage of increasing computation performance by adding additional unitary modules without requiring any modification of the initial architecture or control signals, thus the system is truly modular. Furthermore, embodiments of the invention have the advantage that VLSI technology may be

used to make individual devices which can calculate motion vectors for the various MPEG-2 levels and profiles and for video with any parameters.

A preferred embodiment of the invention may have the advantage that half-pixel precision is achieved using the full anchor frame search by comparing pairs of current frame and anchor frame macroblocks.

An embodiment of the invention will now be described, by way of example only, and with reference to the accompanying drawings, in which:

Figure 1, previously described, shows the movement of a macroblock between a past, present and future frame;

Figure 2, previously described, illustrates half pixel points within a given block of four adjacent pixels;

Figure 3, is a block schematic diagram of the architecture of a motion vector processor embodying the invention;

Figure 4 shows one of the processing elements of figure 3 in greater detail;

Figure 5 is an alternative realisation of the processing element of figure 4 for single pixel precision;

Figure 6 shows, in more detail, one of the parallel pipelined modules P of figure 5;

Figure 7 shows, in more detail, one of the input modules of figure 3;

Figure 8 shows, in more detail, the memory unit of figure 7;

Figure 9 is a block diagram of the Bi module of figure 3;

Figure 10 is a block diagram of the input B module of figure 3;

Figure 11 is a flow chart showing the steps in the anchor frame data priming process for generation of macroblock coordinates;

Figure 12 shows, in more detail, the READ F step in figure 11;

Figure 13 shows, in more detail, the WRITE T step in figure 11;

Figure 14 shows, in more detail, the WRITE F step in figure 11;

Figure 15 is a representation of a anchor frame divided into stripes for processing;

Figure 16 shows an MPEG processor including a motion vector processor embodying the invention;

Figure 17 shows, in block schematic form, the architecture of a multipoint videoconferencing system or a DVD system including the motion vector processor of figure 16.

Figure 18 shows, in block schematic form, the architecture of a videophone system including the motion vector processor of figure 16.

Figure 19 shows, in block schematic form, the architecture of digital video camera including the motion vector processor of figure 16.

Figure 20 shows, in block schematic form, the architecture of television or video encoder including the motion vector processor of figure 16.

The architecture of figure 3 is based on the simultaneous comparison of S current frame macroblocks with K macroblocks of the anchor frame. This may be a portion of the anchor frame or the whole anchor frame depending on the picture size. The macroblocks are preferably 16 x 16 pel luminance pixel blocks although the MPEG 2 standard also supports 16 x 8 luminance pixel blocks or even 8 x 8 chrominance blocks.

It will be appreciated that this approach differs from the prior art in which a single current frame macroblock is compared with the anchor frame macroblocks in the search window. The architecture of figure 3 can be realised on a single VLSI chip but, where K and S are such that a single chip is insufficient, individual modules can be connected together without requiring any reorganisation.

In Figure 3, a plurality of K input modules 20 each receives anchor frame data Ih,Iv on respective inputs 22,24. The output from the Input modules 20(1) to 20(k) is identified as PI(1) to PI(k) and represents a transformed version of the input data. The outputs PI(1) to PI(k) are supplied to a matrix of KxS processing elements 26 identified as PE1.1 to PEk.S in figure 3. Output PI(1) is supplied to the inputs of each of the processing elements in the row PE1.4, that is, elements PE1.1, PE1.2 and PE1.S in figure 3. Output PI(2) is supplied to each of the processing elements in the row PE2.x, that is elements PE2.1, PE2.2 and PE2.S and so on, so that output PI(k) is input to elements Pek.1, Pek.2.... Pek.S as shown in figure 3.

The macroblocks B of the current frame are input on an input 1B to an Input Module 30 which receives them and distributes the current frame macroblocks to S buffers B, shown as 32(1)...32(S) in figure 3. The output of each current frame macroblock buffer B is provided as an input to each processing element in a column.

Thus, buffer B1 provides an input to processing elements PE1.1, PE2.1 and Pek.1 and so on.

The outputs of each of the processing elements PE1.1 to PEk.S are provided as inputs to a row of S comparator modules MIN 1 to MIN S identified by the numeral 34. As with the current frame input buffers 32 the comparators are connected to each processing element in a column of the matrix. Thus, comparator MIN(1) receives at its input the output of processing elements PE1.1, PE2.1 and PEk.1 and so on. The comparators 34 process the inputs to provide X,Y coordinates of matching anchor frame macroblocks for given current frame macroblocks. The X,Y coordinate is the upper left hand coordinate of the block. The comparators then pass this coordinate data to the output block 36.

It will be appreciated that data is input and output serially but all the processing is performed in parallel.

Referring now to figure 4, one of the processing elements 2b is shown in greater detail. The element PEa.b has an input from current frame macroblock buffer Bb and an output to comparator MINb. The element receives comprises four identical parallel-pipelined processing modules 40 shown as Pc, Pv, Ph and PA which each have an output to a comparator MINP 42. Each of the parallel-pipelined processing modules 40 receives as its inputs, the output PB from the column macro block buffer, in this case PBb, and an Input PI from the row Input Module 22.

The Input PI comprises four separate inputs Ic, Iv, Ih and IA which are input respectively to processing modules Pc, Pv, Ph and PA. The processing modules 40 perform parallel comparison of a single macroblock of the current frame provided from buffer B with four interpolations of a macroblock of the anchor frame having coordinates c,v,h and A as defined with reference to figure 2 earlier. Thus, the comparison is made with an anchor block having a given coordinate or coordinates off-set by a half-pixel in a horizontal, vertical or diagonal direction. It is the inclusion of these four pipelined processors in each processing element which gives the ability to estimate motion to half-pixel accuracy.

Figure 5 shows an alternative processing element 26 a.b that is suitable where only a single pixel precision is required. It is identical to the element of figure 4 except that a single parallel pipelined Module 40 is required which receives a single input PI from the input module.

A parallel-pipelined Module 40 is shown in more detail in figure 6. The module comprises M blocks AD 50 operating in parallel, each of which receive as an input the output from the column current frame macroblock buffer together with an Input I. The Input I is provided from the Input Module and will be described in greater detail later. The output of each block AD 50 is passed to an adder-accumulator 60 whose output is the input to processor comparator MIN 42 in figures 4 and 5.

The AD units each carry out a series of arithmetic operations on the incoming data. Thus, the units each include a Subtractor 51 which subtracts the value of the current frame macroblock data from the anchor frame macroblock data, an absolute value Unit 52 which converts the output of the Subtractor to an absolute value, an

accumulating adder 54 which adds the absolute value to the sum of earlier values, a first register 56 which holds the output of the adder 54 and whose output is fed back to the second input to the adder, and a second register 58 which receives the output of the first register 56 and thus the output of the accumulator adder. Thus, the blocks AD calculate the sum of absolute values of M differences with each block performing pipelined operation of sequential devices. The adder accumulator 60 receives the output of each second register 58 of each pipeline as an input to a multiplexer 62. The output of the multiplexer forms the input to an accumulator-adder 64 whose output forms the input to a first register 66 whose output is fed back to adder 54 to provide the second input. Thus, the outputs from the blocks 58 are summed and the output fed to a second register 68 whose output is the input to the comparator MINP 42.

It will now be understood that the comparator MINP 42 of each processing module sequentially compares the sums provided from each of the modules Pc, Pv, Ph, PA for the current frame macroblock and in its most simplistic form, defines with half-pixel precision the coordinates of the anchor frame macroblock which has the smallest partial sum. It will be understood that the macroblock with the smallest partial sum is that which corresponds most closely to the current frame block under consideration. In many applications it will be more appropriate to set a threshold for the comparison. Higher thresholds may be set. As the threshold increases so too does the likelihood that there will be more than one coordinate value which will reach that threshold value. In that case the MPEG 2 standard provides that the decision may be made on the basis either of the first macroblock within the threshold value or the smallest value of all. If a macroblock provides no coordinate value within the threshold, as may be the case, for example, where there is

a scene change, that macroblock is intraframe coded and the remaining macroblocks are interframe coded. This means that the bit rate reduction process is not abandoned purely because one block cannot be matched.

It will be understood that the pipelines AD could be implemented in a variety of other ways.

It will also be understood that it is ideal to process the whole of the frame in parallel but this is not necessary. The amount of the frame that is processed in parallel will depend on the Level/Profile being used and the environment in question. A procedure for optimising the architecture of the processor is described later.

Turning now to figure 7, the Input Module I is shown in more detail. The module comprises the anchor frame buffer 70, shown as Memory Unit I in figure 7 and M processing blocks 72 S1 to Sm together with an adder 74 and a delay line 76. The anchor frame buffer 70 is controlled by a control unit 78.

The purpose of the processing blocks 72 is to provide from the input data the necessary additional data to perform calculations with half pixel precision. Thus, the processing blocks S 72 provide the Ic, Iv, Ih, IA data inputs to the parallel pipelined processing modules 40 of the processing elements. Again it will be understood that if the embodiment of figure 5 is adopted without half-pixel precision, the processing blocks of figure 7 are not necessary.

Referring back to figure 2, four points A,h,v,c are represented in the square. These points are required to operate at half pixel precision. Luminance data Y corresponding to these points is the input to processing modules 40 as mentioned above. Each of the blocks 72

comprises a delay L 80, an adder Sh 82 with delay Lh 84, an adder Sv 86 with delay Lv_1 88 and $Lv_2$90 and adder Sc92. Adder Sh82 performs the horizontal interpolation of equation (2) being half the sum of luminance pixels A+B in figure 2 and thus the delay 84 is of a length equal to the pixel period. The output of adder 82 is the luminance value at point h. Adder Sv performs the vertical interpolation of equation (3) being half the sum of the luminance pixels A+D in figure 2. Adder Sc 92 performs the central interpolation of equation 4 to calculate the luminance at point C in figure 2. Delays L, Lh and Lv_2 all provide timing adjustment for data output on the bus PI. As can be seen from figure 7, the outputs Ic, Iv, Ih and IA are comprised of lines Ic_1, Ic_2... IcM etc, with one line being provided by each of the blocks S1, S2...SM.

Summarising the above, the input module takes the anchor frame data and forms the A,h,v and c data for each of M inputs. The A value is a simple delayed version of the input whereas h,v and c are obtained by performing equations (2),(3) and (4) as described in relation to figure 2.

The additional adder Sv 74 and delay Lv_1 76 shown in figure 7 are required as the value h relative to the last Pixel A to be calculated requires knowledge of the next Pixel B. This is provided by output M+1 from the buffer 70.

Figure 8 shows the input buffer 70 of the input module in more detail. Data inputs Ih, Iv are provided to first and second data registers 100, 102. Data from these registers is transferred to a multiplexer ID4 according to an anchor frame data priming algorithm which will be described. The multiplexer outputs data to a plurality of M+1 two part memory blocks I1 to IM+1 106 which store M+1 columns of anchor frame data. The output of the

multiplexer and the memory blocks 106 are both controlled by signals AR, AWT, AWF from the Control Unit 78 (figure 7). Data is output from the memory blocks to a switch matrix MXI.1-MXI.M+1 108 having M+1 inputs and M+1 outputs. The output of the Switch Matrix is the M+1 lines to the M processing blocks S of figure 7.

The control unit 78 in figure 7 operates according to the anchor frame data priming algorithm and generates the anchor frame macro block coordinates which are sent to the processing elements 26 for processing.

Referring back to figure 3, the current frame macroblock buffer 30 comprises M memory blocks with N cells. The organisation of the buffer 30 enables simultaneous storage of current frame macroblocks and the reading and loading of the next macroblock of the current frame. The memory blocks and registers 32 receive data serially. The organisation of the current frame input buffer is illustrated in figures 9 and 10.

In Figure 10 it will be seen that the input B data is passed to the input B unit register B and a demultiplexer, the output of which passes the data to the Buffers B1 to Bs. As can be seen from Figure 9, each of the B buffers comprises a series of memory blocks 1 to M each having N cells which are duplicated and which blocks have outputs to a respective one of M multiplexers whose outputs are passed to the processing elements of a given column.

The comparator modules 34 MIN1-MINS of figure 3 sequentially compare the partial sums from parts PE1.I to PEk.i and define the coordinates of the anchor frame macroblock for which the threshold criteria are achieved. These coordinates are passed to the output block 36 for output.

Data loading algorithm

Figures 11 to 15 show the steps in the anchor frame priming process to generate the macroblock coordinates. Figure 11 is an overview of the process and figures 12, 13 and 14 show, respectively, the READ F, WRITE T and WRITE F steps in more detail. Figure 15 is a schematic representation of a anchor frame.

Referring first to figure 15, anchor frame 200 having dimensions AxC is divided into K partial cross stripes 202a...k with dimensions Axd, where d=((C-N)/K+N), C is the vertical frame dimension, and K is the number of modules Input I. So, for instance for frame dimensions of 352x288, K=4 and N=16 and the frame is divided into 4 stripes each of dimensions 352x84. The first stripe 202a with upper left angle coordinates (1,1) will be loaded and processed in module Input I1. The second stripe 202b with coordinates (1,68) will be loaded in module Input I2, the third stripe 202c with coordinates (1,136) will be loaded in module Input I3 and the forth one 202k with coordinates (1,204) will be loaded in module Input I4. The stripes are loaded in sequence. All stripes are processed in parallel and in the same manner.

Before the describing the loading algorithm, the following terms will be defined: field F and column T. Field F 204 is part of a stripe that represents number's matrix with the dimensions (M+1)xd. Column T 206 is part of stripe that represents the number's matrix with the dimensions 1xd.

Each of memory modules I1, I2,...,IM+1 (Fig.8) comprises two banks each having a volume d, one of which is using for the processing, the current operational bank,

and the other is used for the loading the next portion of data. Field F is loaded in the bank that currently is used for loading. Each column T of the field F is loaded in the corresponding memory module. This operation is denoted Write F - field load and is shown in figure 14.

The algorithm for the Write F operation provides sequential loading of columns T of field F in corresponding memory modules. In each memory module, column T is loaded sequentially according to the address AWF value.

After the field F is loaded in the first memory bank, the data in this bank is ready for processing. The field F of the next anchor frame will be loaded further in the second memory bank. In the operational memory bank two operations are performed: the field F read operation denoted Read F and the column loading operation denoted Write T. These to operations are illustrated in figures 12 and 13 respectively.

The Read F operation represents the sequence of M+1 simultaneous operand read operations from M+1 memory modules according to the common address ARR. The initial address AR is equal to zero. After N read operations the initial address increments by one and the next N read operations are performed, and so on until the initial address becomes greater then d-N.

After the Read F operation, data of the left column can be replaced with the data of the next to the right side of the field F right column. For instance, after the Write F procedure, the operational memory bank holds

columns data with coordinates Y=1,...,Y=M+1. After the first Read F operation, data of the first column with coordinate Y=1 can be replaced with the data of the next right column with coordinate Y=M+2. This process is performed sequentially for the whole stripe.

Referring to figure 13, the Write T algorithm for the column T loading operates as follows. Firstly the coordinate Y is incremented by one and the new value is compared with the C value (frame vertical dimension). If Y < C, the column loading operation continues. Write address AWT is calculated and then the AWT value is compared with the value of current read address AR (from Read F operation). This comparison is necessary because read and write operations are performed from and to the same memory bank and Read F operation should provide the correct column T data reading. If AWT < AR and there is ready signal from register Rin1 the data is loaded to the address AWT and so on until j<d.

The whole loading algorithm for the loading of one stripe of reference frame is represented at the Fig.11. Firstly field F is loaded in memory through the Write F operation. This operation is synchronized by a ready signal from register Rin2. The finish of this operation is synchronized by the end of loading of S current frame macroblocks in module Input B.

Then for each stripe initial coordinates are set: X=Xf and Y=1. Matrix switch 108 (See Fig.8) provides direct data transfer (MX=0). The address of the column being loaded is set to one: T=1. Then, three parallel processes are being performed in the operational bank: Write F; Read F; and Write T. The last two processes is

synchronized by read address AR. The finish of these processes also is synchronized. If Write T is not outputting signal END Y, then X=Xf, the column loading address is incremented by one (T=(T+1)modM+1), the matrix switch 108 is switched to transfer data according to the column T address (MX=(MX+1)modM+1) and two parallel processes continue to perform until the signal END Y appears. Then the algorithm waits for the finish of field loading (Write F operation) and for the finish of loading of S current frame macroblocks and so on.

In summary the embodiment described provides parallel processing of calculations, anchor and current frame data input and motion vector output through a matrix of processing elements and input modules for the anchor frame and current frame data and an output module for the motion vectors. Motion vectors are calculated in parallel for a set of current frame macroblocks and, preferably, to a half pixel precision. Furthermore, M sums of absolute difference are calculated in parallel in the processing elements and a single macroblock row of 16 pixels is processed in parallel. Pipeline processing is provided for in the calculation of the sum of absolute values of differences, the summing of those sums and the comparison of those sums to determine the closest anchor frame macroblock.

The embodiment has been described with reference to forward predicted coding. It will be appreciated that it is equally applicable to bidirectional coding. The latter is achieved by performing the comparison operation for the current frame twice, once with the forward anchor frame and once with the backward anchor frame and then comparing

the results of the two. The best of the two is then taken as the predicted frame.

It will also be appreciated that the motion estimator can operate on a whole frame of current macroblocks or, where the number of blocks is too high, can process the frame in a number of passes. An alternative would be to use two or more processors, however there is adequate time for at least two passes.

The motion estimator described herein may be used in any environment in which MPEG 2 coding is required. This includes, for example, video signal encoding for broadcast or broadcast quality pictures for subsequent narrowcast or recordal, multipoint tele- or video conferencing equipment, DVD video encoders, video cameras including broadcast quality cameras and camcorders. For applications such as multipoint teleconferencing, it is not practical for the search to be based on a full anchor frame and it is suitable to define a search window. As the amount of movement is likely to be small, it is believed that this approach is satisfactory and can give very significant improvements over presently available systems enabling rates of up to 15 frames per second on conventional ISDN links with a data rate of 128kB. In other applications the statistical approach of the whole frame search is more appropriate. It will be understood that the estimator as described affords the possibility of either solution, depending on the application.

Figures 16 to 20 show examples of how the embodiment of the invention described can be used in a variety of different applications, each using MPEG based video compression.

In figure 16, there is illustrated an MPEG processor 248 which is the core part of all the applications. The MPEG processor comprises a programmable DSP engine 250 to support the basic functions of MPEG video coding and compression and decoding and decompression including DCT, IDCT, Q, Q^{-1}, VL coding and so on. The Motion Detection Processor 252 is a parallel-pipelined processor embodying the present invention. The complexity of the MDP engine 252 will depend on the demands of real-time video sequences being processed for and particular MPEG level and profile. Computational performance of the DSP engine 250 should also be consistent with the particular application.

The MPEG processor proposed can be implemented using existing DSP processors, the example, TMS320C62 DSP processor. Thus it is necessary only to develop the MDP. This two chip solution can be used for the lower MPEG profiles and levels. For higher MPEG levels and profiles it may be necessary to develop a more powerful DSP engine. It is possible to develop a single chip solution for the MPEG processor due to its general structure as outlined above. In the case of a single chip solution, the processor will have one input Data bus and a single interface to the external RAM.

Figure 17 illustrates how an embodiment of the present invention may used in a video conference system. At present, videoconferencing systems are being developed mainly on a PC platform. The embodiment of figure 17 frees the Pentium (or other) PC processor from the hard computational task of determining motion vectors. In figure 17, the system controller 260 communicates with a PCI bus through a PCI interface 262, and with an MPEG processor 264 as illustrated in figure 16 and embodying

the present invention over the system bus. The MPEG processor is coupled to a RAM 266 with which it can exchange data. Depending on the choice of Video and Audio front-end devices 268, 270, and the MPEG processor realisation, the front end devices may either be attached to the system bus (solid line in figure 17), or connected through the system controller 260 (shown as dotted lines in figure 17).

The MPEG processor encodes digital video and audio data from the front end devices. The MPEG data stream is output through the system controller and the PCI bus and can be further transported to the destination through the communications capabilities of the PC.

The MPEG processor 262 also decodes incoming audio and video data which is received as an MPEG data stream on the PCI bus. Decompressed audio and video data is further available to the user through the PCI bus and the corresponding PC capabilities such as the monitor and sound blaster.

The use of a PC or other computer with the MPEG acceleration board will allow multi-point videoconferencing systems to be built by using the computational resources of existing processors such as the Pentium and Pentium II (TM) to decode additional input MPEG channels.

The system outlined above is suitable for a number of videoconferencing systems such as point-to -point QCIF videoconferencing, multipoint QCIF videoconferencing and low-bit CIF videoconferencing on ISDN lines. The

processing of audio data is optional and may be performed using PC software or by the DSP engine.

A DVD system embodying the present invention has the same architecture as shown in figure 17. Differences may exist in the MPEG processor due to the need to compression conforming to CCIR Rec 601 standard. To provide the corresponding MPEG level and profile, a more complex MDP engine is required. As the system is intended only to compress video and audio and to write an MPEG stream on DVD ROM through the PCS capabilities, the increase in DSP complexity, if any may be negligible.

Figure 18 shows how an MPEG processor embodying the present invention and as shown in figure 16 may be used in a videophone system. The system is based on a QCIF videoconferencing system and is similar to the system illustrated in figure 17 except that it requires audio and video back end devices 272, 274 which provide digital to analog conversion of decompressed MPEG data. In addition the system controller interface must include a modem interface 276 for exchange of digital MPEG data between the transmitting and receiving points. In this system, audio data processing is necessary.

Figure 19 shows how an MPEG processor embodying the present invention and illustrated in figure 16 may be used, in conjunction with DVD technology for MPEG data storage to develop a digital video camera. This realisation relies on the availability of rewritable DVD-ROMs with sufficiently good speed characteristics. The arrangement is similar to that of figure 18 except that the audio and video back end devices are optional if play

back is required, that a DVD controller 278 communicates with the system controller and that no modem is needed.

Figure 20 shows an example of how the MPEG encoder embodying the invention and illustrated in figure 16 may be used as a television MPEG encoder. The circuit illustrated may be used in broadcasting equipment to encode a single television channel. The same configurations may be used for standard definition and HDTV with the difference being in the complexity of the MPEG processors. Present fabrication techniques can build a processor for standard definition on a single chip. At present several chips operating in parallel are required to support HDTV although it is envisaged the a single chip solution will be possible shortly as fabrication techniques improve.

Procedure for the architecture parameters definition

It will be appreciated that for different MPEG levels and profiles and for different applications, varying sizes of processing matrix will be required. The following section sets out how the parameters of the architecture may be defined.

For real-time motion vectors calculations it is necessary to define the following architecture parameters:

Number of processing elements K*S;

Number of input ports for module Input B - LB;

Number of input ports for module Input I - LIv and LIh;

Number of memory modules cells - D;

Number of output ports for module Output V - LV;

Number of the processing elements in horizontal direction - S;

Number of the processing elements in vertical direction - K.

The architecture parameters depend on the values of the following primary data:

A - frame horizontal dimension;
C - frame vertical dimension;
p - number of bits for pixel presentation;
MxN - macroblock dimensions;
Tc - time interval for single operation on pixel in the pipeline and memory read time interval;
Tio - time interval for the external single information bit input/output;
T - time interval for the calculation of the motion vectors for the full current frame;
Lmax - maximal number of input/output ports.

Calculation of K*S value

Calculation of K*S matrix dimensions necessary for real-time operation, that is the total number of processing elements is based on the following expression:

$$T=(A/M*C/N*(A-M)*(C-N)*N*Tc)/K*S \quad (1).$$

This expression means that during time interval T it is necessary to perform block matching procedures on A/M*C/N current frame macroblocks with (A-M)*(C-N) anchor frame macroblocks using matrix of K*S parallel processing elements. Block matching procedure for two macroblocks requires time interval N*Tc as the only read operation of N macroblock rows is performed sequentially and all other necessary operations for the block matching procedure are performing in parallel-pipelined mode.

Therefore, value of K*S is calculating from expression (1):

$$K*S \geq (A*C*(A-M)*(C-N)/M)*(Tc/T) \quad (2).$$

Maximal value of S is defining by the number of anchor frame macroblocks:

S_{max}= A/M*C/N

(3).

Minimal value of K in this case is:

K_{min}=K*S/S_{max}=(A-M)*(C-N)*N*(Tc/T)

(4).

Calculation of Input/Output ports number

Value of LB is calculating from the following expression:

T≥A*C*p*(Tio/LB)

(5).

This expression means that during time interval T it is necessary to perform the loading of whole current frame data into processor. So, the value of LB is:

LB≥A*C*p*(Tio/T)

(6).

Value of LV is calculating from the following expression:

T≥2*(A/M)*(C/N)*$\log_2$ Å*(Òio/LV)

(7).

This expression means that during time interval T it is necessary to output X,Y coordinates of all calculated motion vectors for current frame. So, the value of LV is:

LV≥2*(A/M)*(C/N)*$\log_2$ A*(Tio/T)

(8).

Value of LIv is calculating from the following expression:

T≥ (M+1)*p* A*C^2/S*M*N*(Òio/LIv)

(9).

This expression means that during time interval T it is necessary to load memory volume equal to (M+1)*p*C by (A*C)/(S*M*N) times. So, the value of LIv is:

LIv≥ (M+1)*p* A*C²/S*M*N*(Òio/T)

(10).

Value of LIh is calculating from the following expression:

T≥ ((À-Ì-1)*ð*Ñ²*A)/(S*M*N)*(Òio/LIh)

(11).

This expression means that during time interval T it is necessary to load memory volume equal to (A-M-1)*p*C by (A*C)/(S*M*N) times. So, the value of LIh is:

LIh≥ ((À-Ì-1)*ð*Ñ²*A)/(S*M*N)*(ÒiO/T)

(12).

Calculating of D value

D is the length of column that is loading in K Input I modules. The D value could be calculated from the following expression:

D*p*(Tio/LIh)≤(D-N)/K*N*Tc

(13).

This expression means that during time interval for loading the column with the length equal to D it is necessary to load (D-N)/K*N operands. Therefore, D is calculating from the following expression:

D≥ N/(1-(p*Tio/N*Tc)*(K/LIh))

(14).

Using the expression (12) for the Lih final expression for D value is:

D≥ C*(1-1/(A-M))/(1-1/(A-M)*C/N)

(15).

Since the value of (1-1/(A-M))/(1-1/(A-M)*C/N) is greater 1, D>C and it is in contradiction with the loading algorithm. So, it is necessary to choice D=C. In this case the expression (12) will be:

LIh≥ ((À-Ì)*ð*Ñ²*A)/(S*M*N)*(Òio/T) (16).

From expressions (10) and (16) it is possible to define S_{min} with the restrictions on the Input/Output ports number:

S_{min}=(((A+1)*A*C²*p)/(M*N))*(Tio/T)*(1/Lmax-LB-LV) (17).

Calculation of K and S

By increasing the S value it is possible to decrease the number of Input/Output ports. On the other hand the increasing of K value could lead to the decreasing of processor's hardware.

Suppose:

H - necessary hardware for the PE implementation according to the Fig. 4;

gs*H - necessary hardware for the Input B module implementation;

gk*H - necessary hardware for the Input I module implementation.

Coefficients gs and gk depend on the particular modules implementations.

So, the total hardware for the implementation of processor for the motion vectors calculations according to Fig. 3 could be minimized using the following expression:

K*S+K*S/K*gs*H+K*gk*H→min (18).

In order to minimize hardware it is necessary to differentiate by K expression (18) and to equate result to zero. In this case the optimal value K_{opt} equal to:

$K_{opt}= (K*S*gs/gk)^{1/2}=((A*C*(A-M)*(C-N)/M)*(Tc/T)*gs/gk)^{1/2}$ (19).

If $K_{opt} > K_{max}$ then $K=K_{opt}$; otherwise $K=K_{max}$ and S is calculating from (2):

$$S=(A*C*(A-M)*(C-N)/M)*(Tc/T)/K \quad (20).$$

In the case of Input/Output ports number restrictions applying further it is necessary to perform the following final calculations:

If $S \geq S_{min}$ then $S=S$; otherwise $S=S_{min}$ and K should be recalculated from (2):

$$K=(A*C*(A-M)*(C-N)/M)*(Tc/T)/S \quad (21).$$

Table 2 below represents the results of applying of the optimization procedures for various video formats. In all calculations the following initial parameters were used:

p=8;
N=16;
M=16;
T=0.0166 sec;
gs=0.2;
gk=1.2;
D=C.

Table 2

A25	N	Tc,ns	Tio,ns	K*S	LB	LV	LIv	LIh	L	K_{opt}	S
352	288	1	10	35	1	1	1	16	19	3	12
720	576	0.4	4	252	1	1	1	36	39	7	36
1440	1152	0.2	2	2021	2	1	2	76	81	19	76
1920	1152	0.2	2	3603	3	1	3	126	133	25	126

Variations and modifications to embodiments described are possible without departing from the invention and will occur to those stated in the art. However, the invention defined solely by the claims appended hereto.

CLAIMS

1. A parallel processor for estimating motion of a given portion of a current image frame with reference to a anchor frame comprising:

an input for receiving current frame data;

an input for receiving anchor frame data;

a two-dimensional matrix of rows and columns processing elements each processing element for comparing a given area of the current frame with at least an area of the anchor frame wherein the matrix simultaneously compares S areas of the current frame with nK areas of the anchor frame, each column of processing elements simultaneously comparing one field of the current frame data with nK fields of the anchor frame, and each row of processing elements simultaneously comparing S fields of the current frame data, with n fields of the anchor frame, the matrix having dimensions of KxS and n being an integer;

means for selecting from the comparison, for each area of the current frame, an area of the anchor frame corresponding to the area of the current frame; and

means for outputting data identifying the selected areas of the anchor frame.

2. A parallel processor according to claim 1, wherein the matrix simultaneously compares S areas of the current frame with 4K areas of the anchor frame.

3. A parallel processor according to claim 1 or 2 wherein the areas of the anchor frame and the current frame are all cosized macroblocks.

4. A parallel processor according to claim 3, wherein the macroblocks comprise 16x16 pixels.

5. A parallel processor according to claim 4, wherein the pixels are luminance pixels.

6. A parallel processor according to any preceding claim, wherein each processing element comprises a comparator and at least one parallel pipeline processor, wherein the at least one parallel pipeline processor receives current frame image area data and anchor frame image area data and outputs a sum of absolute differences between the current frame image area data and the anchor frame image area data to the comparator.

7. A parallel processor according to claim 6, wherein the parallel pipeline processor comprises a plurality of pipeline stages arranged in parallel and operating simultaneously and a pipeline accumulating adder for adding the outputs of each of the pipeline stages.

8. A parallel processor according to claim 7, wherein each of the pipeline stages comprises a subtractor for providing a differential output from anchor and current frame data inputs, an absolute value calculator, an accumulator adder for adding calculated absolute values and first and second registers for holding the accumulated absolute values.

9. A parallel processor according to claim 8, wherein the pipeline accumulating adder sums the outputs of the second registers of each pipeline stage.

10. A parallel processor according to claim 7, 8 or 9, wherein the accumulating adder comprises a multiplexer for receiving data inputs from the pipeline stages, an adder for summing data inputs, a first register for holding the output of the adder, wherein the adder receives as a further input the content of the register, and a further

register for receiving the output of the first register for output to the comparator of the processing element.

11. A parallel processor according to any of claims 6 to 10, wherein each processing element comprises four parallel pipeline processors, the outputs of which are input to the comparator, wherein the four parallel pipeline processors perform parallel comparison of a single area of the current frame with four areas of the anchor frame separated for full pixels and half pixels obtained by horizontal, vertical and/or diagonal interpellation.

12. A parallel processor according to any preceding claim, wherein the anchor frame data input comprises a anchor frame buffer and a plurality of parallel processing blocks for processing simultaneously pixels of a row of the frame area, and a control unit.

13. A parallel processor according to claim 12, wherein each parallel processing block comprises a first means for generating a value of a pixel at a position offset horizontally half a pixel from an input pixel position, a second means for generating a value of a pixel at a position offset vertically half a pixel from said input pixel position, and a third means for generating a value of a pixel at a position offset vertically and horizontally half a pixel from said input pixel position.

14. A parallel processor according to claim 13, wherein said first means comprises an adder and a first delay means and performs the function $h = (A+B)/2$ where h is the half pixel offset value and A and B are horizontally adjacent input pixels.

15. A parallel processor according to claims 13 or 14, wherein said second means comprises an adder and a delay

means and performs the function v = (A+D)/2 where v is the half pixel offset value and A and D are vertically adjacent input pixels.

16. A parallel processor according to claims 13, 14 or 15, wherein said third means comprises an adder and performs the function c = (A+B+D+E)/4 where C is the value of the offset pixel and A,B,D and E are horizontally and vertically adjacent pixels.

17. A parallel processor according to any of claims 12 to 16, wherein the control unit controls the anchor frame buffer and outputs reference area coordinate values to the processing elements.

18. A parallel processor according to any of claims 12 to 17, wherein the anchor frame buffer comprises M+1 memory blocks, where M is a dimension of the anchor frame area, and a switch matrix receiving data input from the memory blocks.

19. A parallel processor according to any preceding claim, wherein the anchor frame data input comprises an input module for receiving and distributing input data, and S current frame area buffers which receive current frame area data from the input module.

20. A parallel processor according to claim 19, wherein the input module comprises a plurality of memory blocks, each block comprising a pair of memory banks each having a plurality of memory cells.

21. A parallel processor according to claim 20, wherein the number of memory blocks in the anchor data input module is M and the number of cells in each memory block is N where MxN is the dimension of the current frame area.

22. A parallel processor according to any preceding claim, wherein the selecting means comprises S comparators, said comparators also defining the coordinates of the anchor frame areas corresponding to a given current frame area.

23. A parallel processor according to any preceding claim wherein n is 1 or 4.

24. A video processor comprising a programmable DSP engine and a motion detection processor, wherein the motion detection processor comprises a parallel processor according to any preceding claim.

25. A video processor according to claim 24, wherein the video processor is an MPEG processor.

26. A video encoder comprising a video processor according to claim 24 or 25.

27. A video encoder according to claim 26, further comprising a system controller communicating with the video processor, a random access memory communicating with the video processor, an audio front end and a video front end, wherein the audio and video front ends communicates with the video processor either via the system bus or via the controller.

28. A multipoint teleconferencing apparatus comprising a video processor according to claim 24 or 25.

29. A multipoint teleconferencing apparatus according to claim 28, further comprising a system controller communicating with the video processor, a further processor, an interface between the system controller and the processor, a random access memory communicating with the video processor, and a video front end, wherein the video front end communicates with the video processor either via the system bus or via the controller.

30. A multipoint teleconferencing apparatus according to claim 29, wherein the further processor is a PC.

31. A DVD system comprising a video processor according to claim 24 or 25.

32. A DVD system according to claim 31, further comprising a system controller communicating with the video processor, a further processor, an interface between the system controller and the processor a random access memory communicating with the video processor,and a video front end, wherein the video front end communicates with the video processor either via the system bus or via the controller.

33. A DVD system according to claim 32, wherein the further processor is a PC.

34. A digital videophone system comprising a video processor according to claim 24 or 25.

35. A digital videophone system according to claim 33, further comprising a system controller communicating with the video processor, a modem interface communicating with the system controller, a random access memory communicating with the video processor, a video front end, an audio front end wherein the video front end communicates with the video processor either via the system bus or via the controller, an audio back end and a video back end, wherein the aduio and video back ends are connected to the system controller.

36. A digital video camera comprising a video processor according to claim 24 or 25.

37. A digital video camera according to claim 36, further comprising a system controller communicating with the video processor, a random access memory communicating with the video processor, an audio front end, a video front end, wherein the audio and video front ends communicates with the video processor either via the system bus or via the controller, and a DVD controller.

38. A procedure for defining the architectural parameters of a parallel processor for estimating motion according to any of claims 1 to 23, comprising the steps of:

calculating the number of processing elements K*S where K*S is a matrix having K columns and S rows, comprising determining the time period required to perform block matching procedures on current frame macroblocks;

calculating the number of input/output ports from the time interval required to load current frame data and anchor frame data for procesing and to output coordinate data of calculted motion vectors;

calculating the size of memory cells required to enable a column of inputs to be loaded in a given time; and

calculating the number of processing elements in both the vertical and horizontal directions by calculating the optimum value of K based on the processor hardware necessary to implement the processor.

PARALLEL PROCESSOR FOR MOTION ESTIMATOR

Abstract

A parallel processor for estimating motion between macroblocks of a current frame and an anchor frame comprises a KsS matrix of processing elements (26), K input modules (20) each for inputting anchor frame data to a row of processing elements, an input module (30) for inputting current frame data, S current frame buffers each for inputting current frame macroblock data to each of a column of processing elements, S comparator modules each for comparing the output of each of a column of processing elements, and an output module fo routputting coordinates of anchor frame macroblocks most similar to given current frame macroblocks. The S current frame macroblocks may each be compared simultaneously compared with nK anchor frame macroblocks thereby significantly reducing processing time.

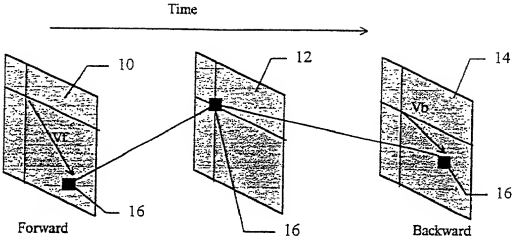

Fig. 1

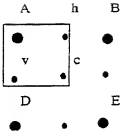

Fig. 2

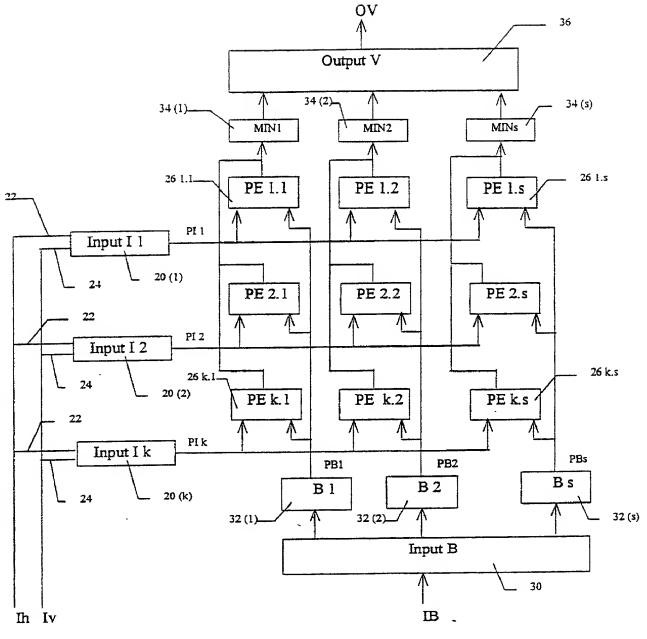
OV
36
Output V
34 (1)
34 (2)
34 (s)
MIN1
MIN2
MINs
26 1.1
26 1.s
PE 1.1
PE 1.2
PE 1.s
22
Input I 1
PI 1
20 (1)
24
PE 2.1
PE 2.2
PE 2.s
22
Input I 2
PI 2
24
20 (2)
26 k.1
26 k.s
PE k.1
PE k.2
PE k.s
22
Input I k
PI k
24
20 (k)
PB1
PB2
PBs
B 1
B 2
B s
32 (1)
32 (2)
32 (s)
Input B
30
Ih
Iv
IB

Fig. 3

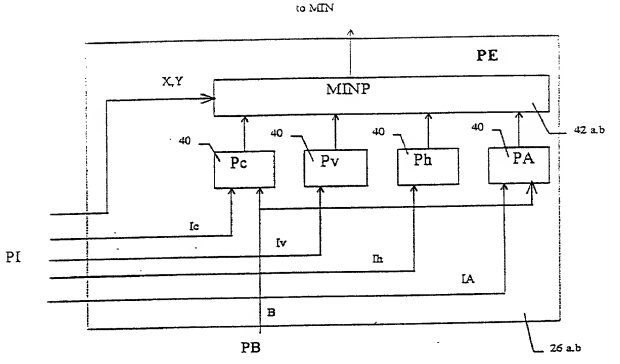
to MIN
PE
X,Y
MINP
42 a,b
40
40
40
40
Pc
Pv
Ph
PA
Ic
Iv
Ih
IA
PI
B
PB
26 a,b

Fig. 4

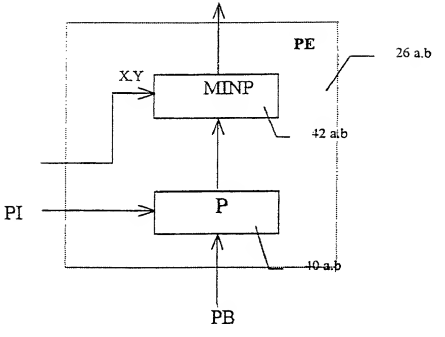

Fig. 5

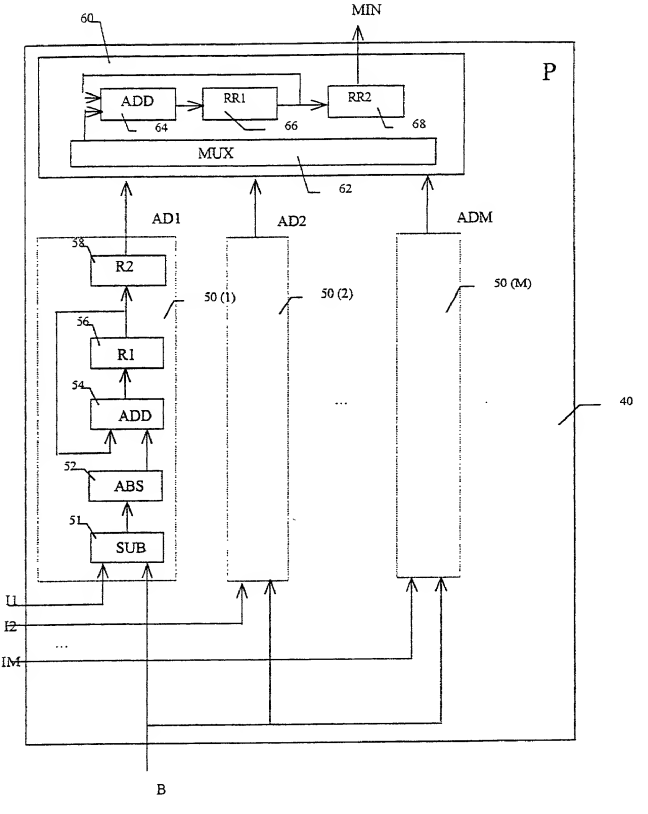
60
MIN
P
ADD
RR1
RR2
64
66
68
MUX
62
AD1
AD2
ADM
58
R2
50 (1)
50 (2)
50 (M)
56
R1
54
ADD
40
52
ABS
51
SUB
I1
I2
...
IM
B

Fig. 6

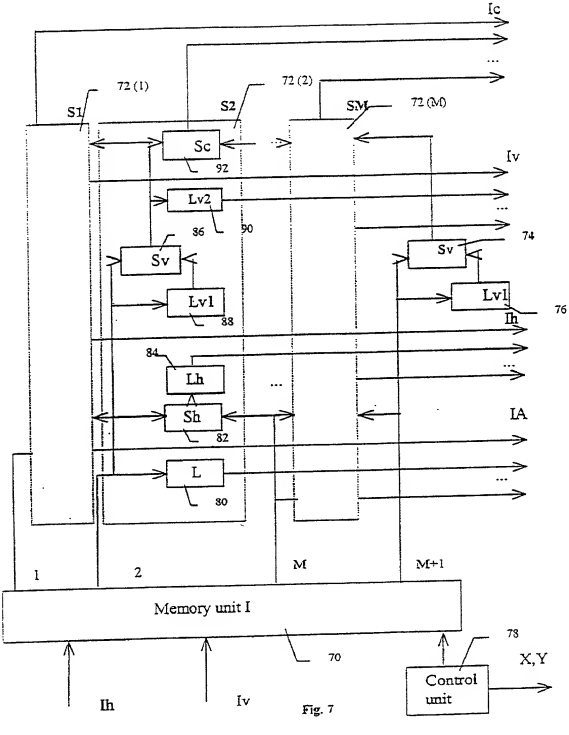
Ic
72 (1)
72 (2)
72 (M)
S1
S2
SM
Sc
92
Iv
Lv2
86
90
Sv
74
Lv1
88
Ih
76
84
Lh
Sh
82
IA
L
80
1
2
M
M+1
Memory unit I
70
78
X,Y
Control unit
Ih
Iv

Fig. 7

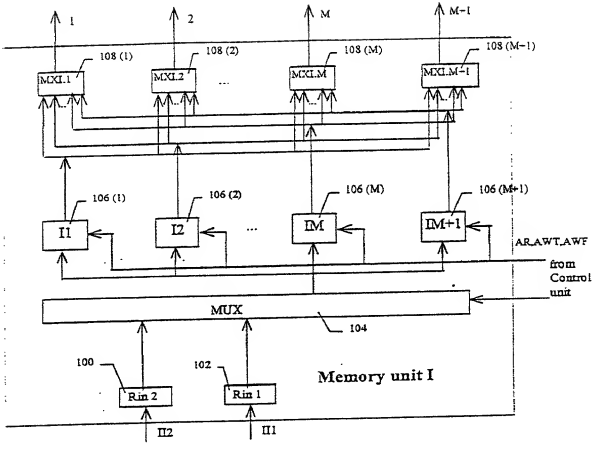
1
2
M
M-1
108 (1)
108 (2)
108 (M)
108 (M-1)
MXI.1
MXI.2
MXI.M
MXI.M-1
106 (1)
106 (2)
106 (M)
106 (M+1)
I1
I2
IM
IM+1
AR_AWT_AWF
from
Control
unit
MUX
104
100
102
Rin 2
Rin 1
Memory unit I
II2
II1

Fig. 8

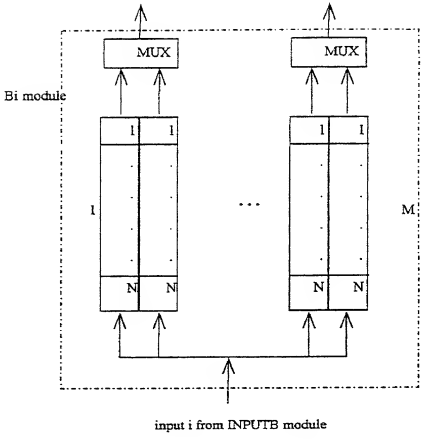

Fig. 9

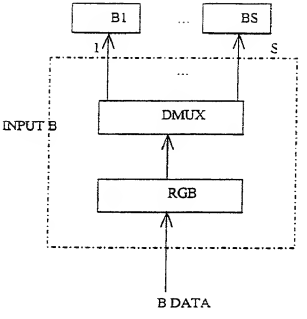

Fig. 10

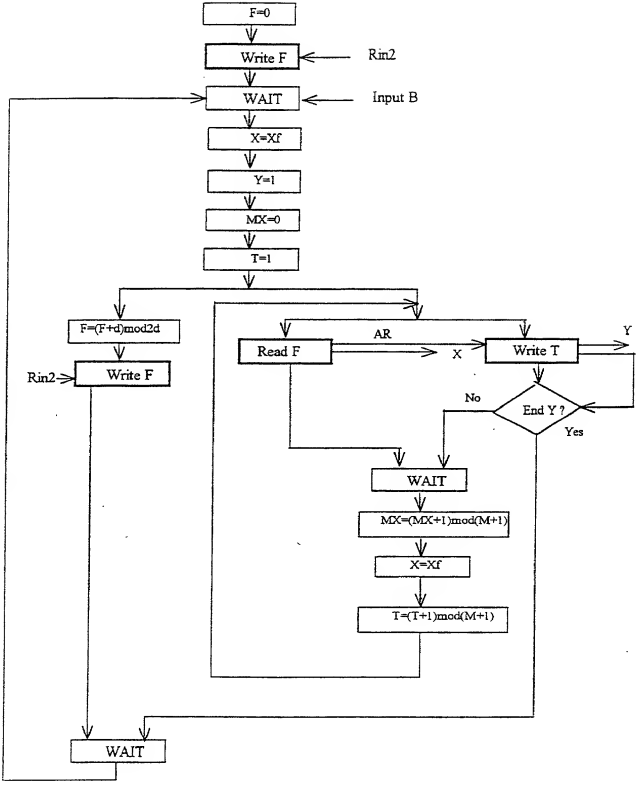
F=0
Write F
Rin2
WAIT
Input B
X=Xf
Y=1
MX=0
T=1
F=(F+d)mod2d
Rin2
Write F
Read F
AR
X
Write T
Y
No
End Y ?
Yes
WAIT
MX=(MX+1)mod(M+1)
X=Xf
T=(T+1)mod(M+1)
WAIT

Fig. 11

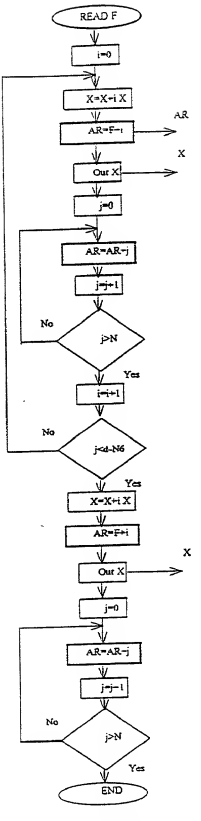
READ F
i=0
X=X+i X
AR=F-i
AR
Out X
X
j=0
AR=AR-j
j=j+1
No
j>N
Yes
i=i+1
No
j<d-N6
Yes
X=X+i X
AR=F+i
Out X
X
j=0
AR=AR-j
j=j-1
No
j>N
Yes
END

Fig. 12
12/20

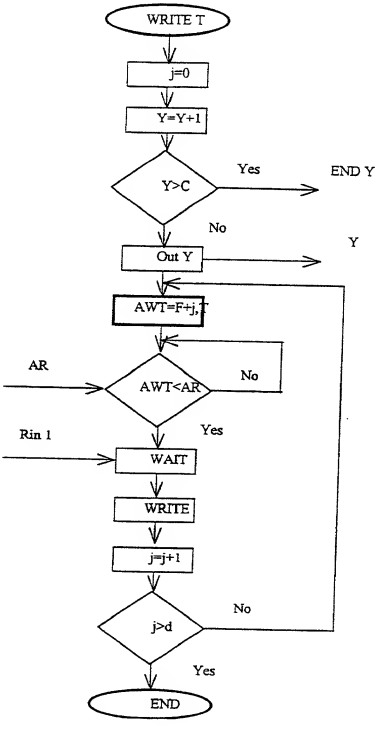
WRITE T
j=0
Y=Y+1
Y>C
Yes
END Y
No
Out Y
Y
AWT=F+j,T
AR
AWT<AR
No
Rin 1
Yes
WAIT
WRITE
j=j+1
j>d
No
Yes
END

Fig. 13

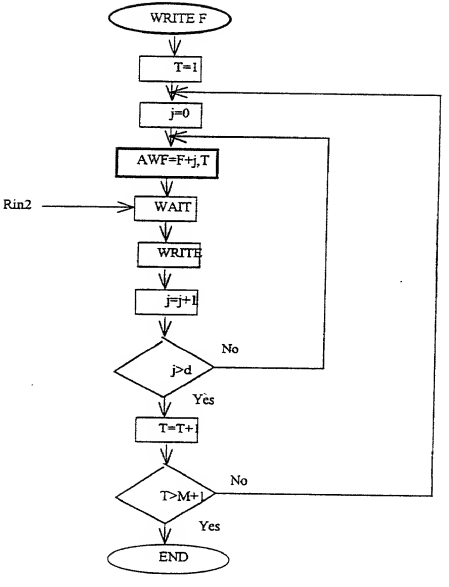
WRITE F
T=1
j=0
AWF=F+j,T
Rin2
WAIT
WRITE
j=j+1
No
j>d
Yes
T=T+1
No
T>M+1
Yes
END

Fig. 14

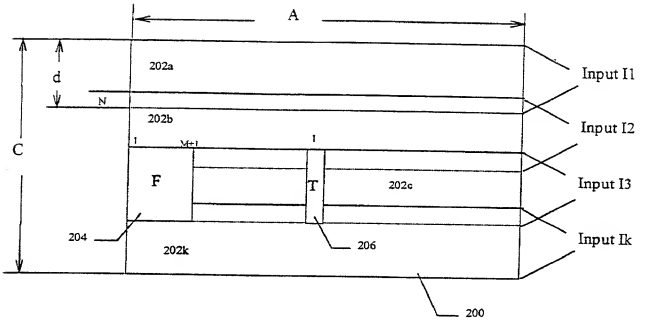
A
d
C
N
202a
202b
1
M+1
I
F
T
202c
202k
204
206
200
Input I1
Input I2
Input I3
Input Ik

Fig. 15

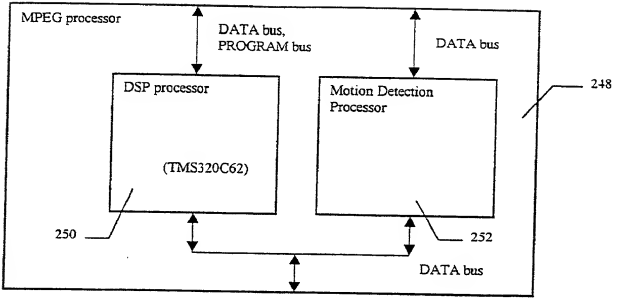
MPEG processor
DATA bus, PROGRAM bus
DATA bus
248
DSP processor
(TMS320C62)
Motion Detection Processor
250
252
DATA bus

Fig. 16

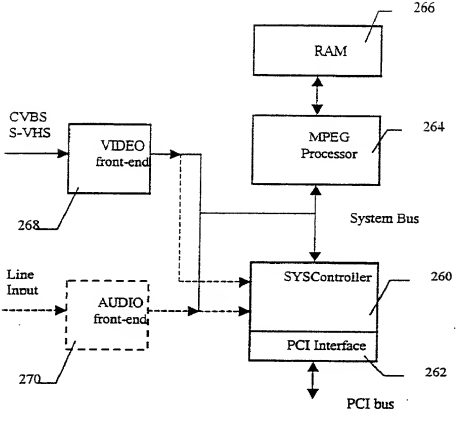
266
RAM
CVBS
S-VHS
VIDEO
front-end
MPEG
Processor
264
268
System Bus
Line
Input
AUDIO
front-end
SYSController
260
PCI Interface
270
262
PCI bus

Fig. 17

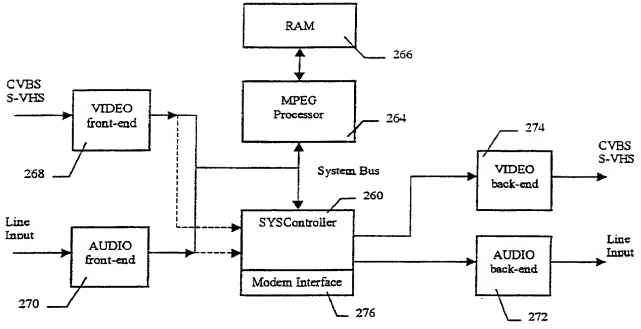
RAM
266
CVBS
S-VHS
VIDEO
front-end
MPEG
Processor
264
274
CVBS
S-VHS
VIDEO
back-end
268
System Bus
260
SYSController
Line
Input
AUDIO
front-end
AUDIO
back-end
Line
Input
Modem Interface
270
276
272

Fig. 18

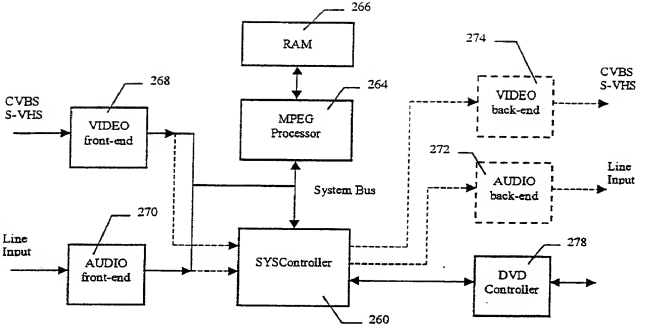
266
RAM
274
CVBS
S-VHS
268
264
CVBS
S-VHS
VIDEO
front-end
MPEG
Processor
VIDEO
back-end
272
AUDIO
back-end
Line
Input
System Bus
270
Line
Input
AUDIO
front-end
SYSController
278
DVD
Controller
260

Fig. 19

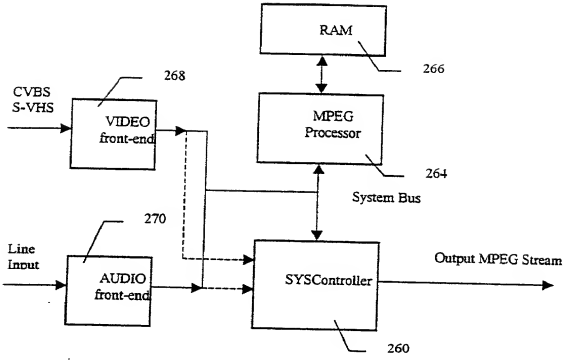
RAM
266
268
CVBS
S-VHS
VIDEO
front-end
MPEG
Processor
264
System Bus
270
Line
Input
AUDIO
front-end
SYSController
Output MPEG Stream
260

Fig. 20

Attorney Docket No.: 13109/002001/39804US

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention PARALLEL PROCESSOR FOR MOTION ESTIMATOR, the specification of which:

[ ] is attached hereto.
[X] was filed on 6/19/01 as Application Serial No. 09/830,049 and was amended on _____.
[ ] was described and claimed in PCT International Application No. _____ filed on _____ and as amended under PCT Article 19 on _____.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose all information I know to be material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim the benefit under Title 35, United States Code, §119(e)(1) of any United States provisional application(s) listed below:

U.S. Serial No.	Filing Date	Status

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose all information I know to be material to patentability as defined in Title 37, Code of Federal Regulations, §1.56(a) which became available between the filing date of the prior application and the national or PCT international filing date of this application:

U.S. Serial No.	Filing Date	Status

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed:

Country	Application No.	Filing Date	Priority Claimed
WO	PCT/GB99/03458	10/18/1999	[X] Yes [ ] No
GB	9822799.3	10/15/1998	[X] Yes [ ] No

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Attorney Docket No.: 03109/002001/998oUS

Scott C. Harris, Reg. No. 32,030
David L. Feigenbaum, Reg. No. 30,378
Hans R. Troesch, Reg. No.36,950
Bing Ai, Reg. No. 43,312
Frederick H. Rabin, Reg. No. 24,488
John C. Phillips, Reg. No. 35,322

William J. Egan, III, Reg. No. 28,411
Dorothy Whelan, Reg. 33,814
James T. Hagler, Reg. 40,637
Richard J. Anderson, Reg. No. 36,232
Samuel Borodach, Reg. No. 38,388
Kenyon Sencher, Reg. No. 41,873

12

Address all telephone calls to SCOTT C. HARRIS at telephone number (858) 678-5070 ext 4321.

Address all correspondence to SCOTT C. HARRIS at:

FISH & RICHARDSON P.C.
Customer Number: 20985
4350 La Jolla Village Drive, Suite 500
San Diego, CA 92122

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Full Name of Inventor:	SERGGEY ARTAMONOV
Inventor's Signature:	Date: 20/07/2001
Residence Address:	62-351, K-427 Moscow, 103427, Russia RUX
Citizenship:	Russian
Post Office Address:	same as above

Full Name of Inventor:	VLADIMIR KOZLOV
Inventor's Signature:	Date: 20/07/2001
Residence Address:	14-337, K-482 Moscow, 103482, Russia RUX
Citizenship:	Russian
Post Office Address:	same as above

10

2a

Attorney Docket No.: 13109/002001/39504US

ull Name of Inventor:	YURY ZAULIVETER
nventor's Signature:	Date: 20/07/2001
esidence Address:	14-59-1-20 Kirovogradskaya Street, Moscow, 112587, Russia RUX
itizenship:	Russian
ost Office Address:	same as above

ull Name of Inventor:	ELENA FISCHENKO
nventor's Signature:	Date: 20/07/2001
esidence Address:	201-26 Lieninskaya Street, 115404, Russia RUX
itizenship:	Russian
ost Office Address:	same as above